Software Quality Assurance

MiniSIS

INSE 6260 Software Quality Assurance

Final Report

Submitted To:

Rachida Dssouli

Submitted By:

Nidhi Arora(40014504)

Karun Sharma(40016964)

Software Quality Assurance

MiniSIS

Table of Contents

1. System Architecture

1.a. Overview

The MiniSIS (Student Information System) is a standalone desktop application meant for course Co-Ordinator, Instructors, Program Director and Students. This MiniSIS's features will be based on users' responsibilities and their duties. It will allow the students to login, register to courses by adding or dropping the courses, view cumulative GPA, and get their transcript.

Considering the rules of the University (taken as a reference), it facilitates the assistance in the course registration and getting transcript with the help of course coordinator of respective department. Furthermore, instructor will publish the student grade to get the CGPA. Student and instructor can check the schedule of course. The administrator will have specific roles to manage the MiniSIS such as managing the login process of all stakeholders.

The system also allows administrator to add term, course detail, user, and pre-requisite courses (for students that require to do that pre-requisite course to the system).Login system allows user to recover their password in case they forget. Authentication for that is done using security questions. Administrator is given the authority to reset password for all the users if needed. The Program Director has the authority of waiving off the pre-requisite courses of students.

1.b. Architecture

A hybrid of MVC and three-tier architecture is used for building MiniSIS as shown in Figure 1.In MVC, the View is what is rendered on the screen of the GUI. How it is rendered is where the software part comes in. The Model has the business logic or software that collects and retrieves any data required for the View. The Controller responds to user events that trigger changes or activities from both the View and the Model. The MVC pattern does not describe how to best design the data access and how to manage the complexity that can occur within the business and data tiers. As the internet, has become more open and further entrenched in business processes, the demand to access the business tier without the presentation tier has increased. The creation of Business Tier modules allows for other approaches to access business logic from other sources, enabling business-to-business interactions and innovations. [1].Figure 1 shows the MVC and three tier architecture that has been adopted.

Figure 2 shows the package structure used in MiniSIS that corresponds to MVC architecture with model at the business layer and database as third layer in the tier.

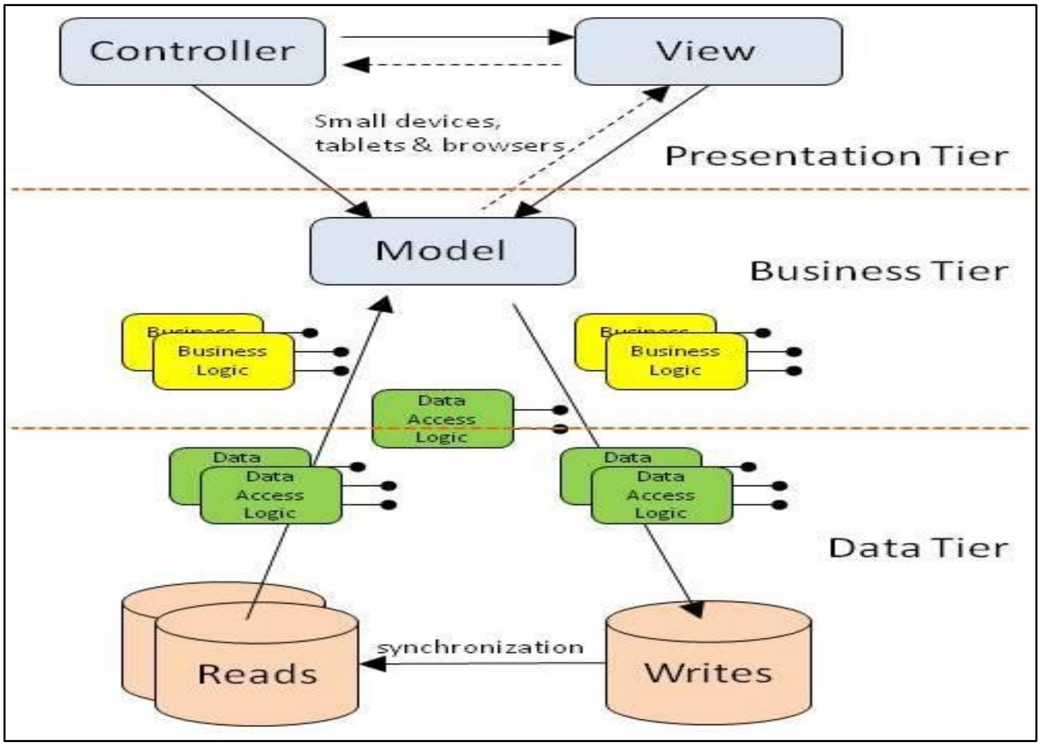


Figure 1 System Architecture

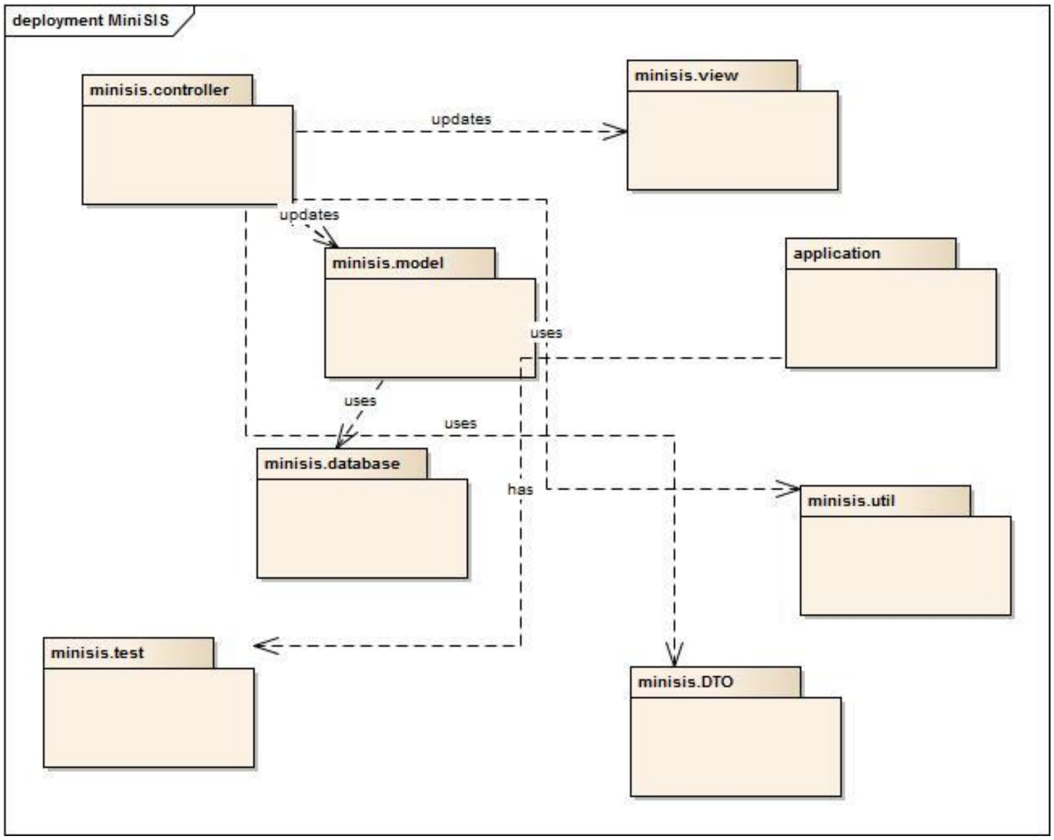


Figure 2 System Architecture in MiniSIS

1.c. Database Design

There are 18 tables designed in the database. The tables are in normalized form. There are no duplicate entries in the database. “admin” and “student” tables are used to store the information of admins(Course Coordinator and Program Director) and students respectively. There is a table named as “users” The table “registration” records the relationship of students and the courses they choose “security questions” table is added to ease the login. For scheduling, “building”, “room”, data of which is used to schedule the courses. A table “pre-requisite” is added to create functionality of waiving off course which is provided to the Program Director.

Figure 3 and 4 show the schema and ERD of MiniSIS respectively.

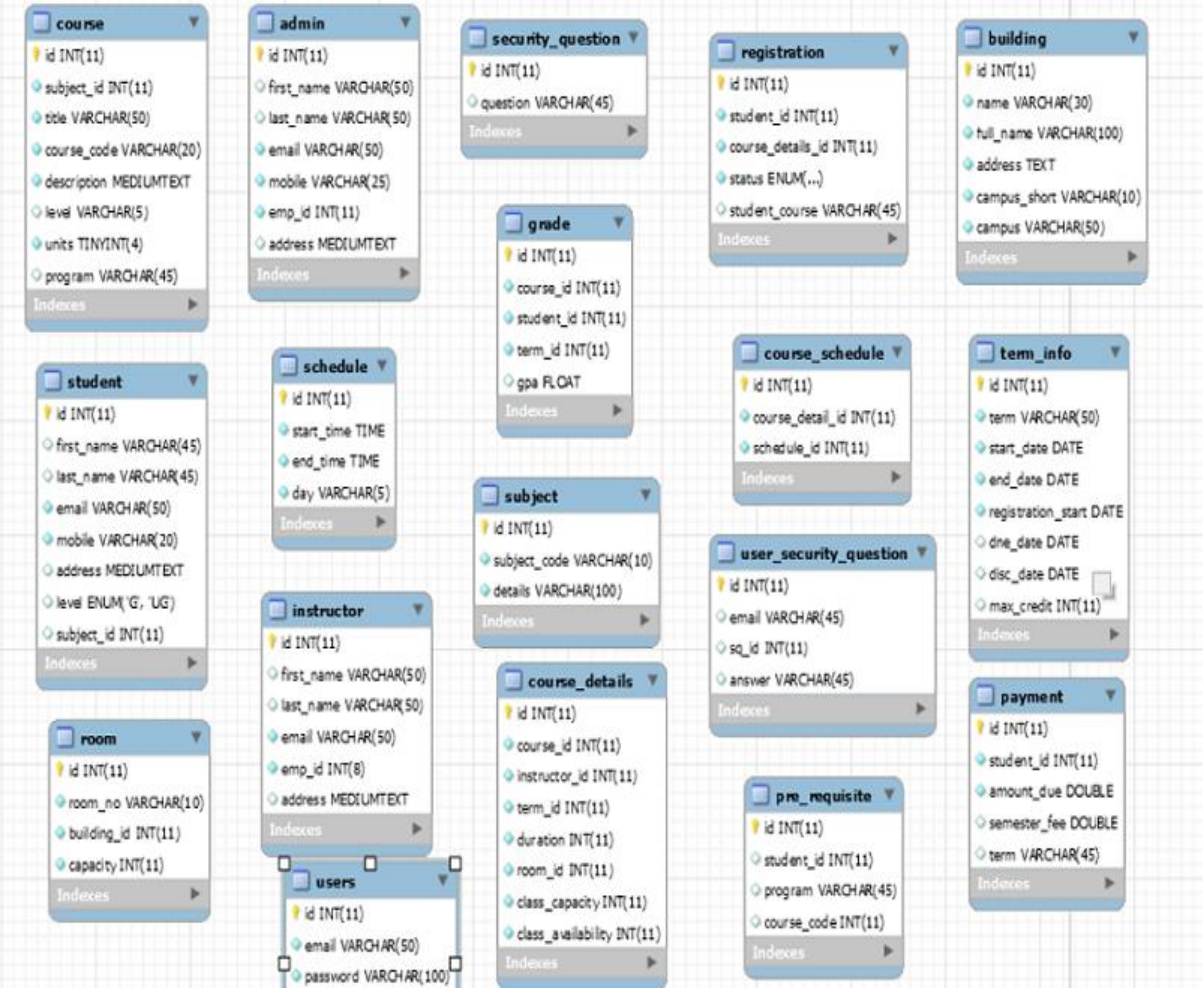


Figure 3 Schema of MiniSIS

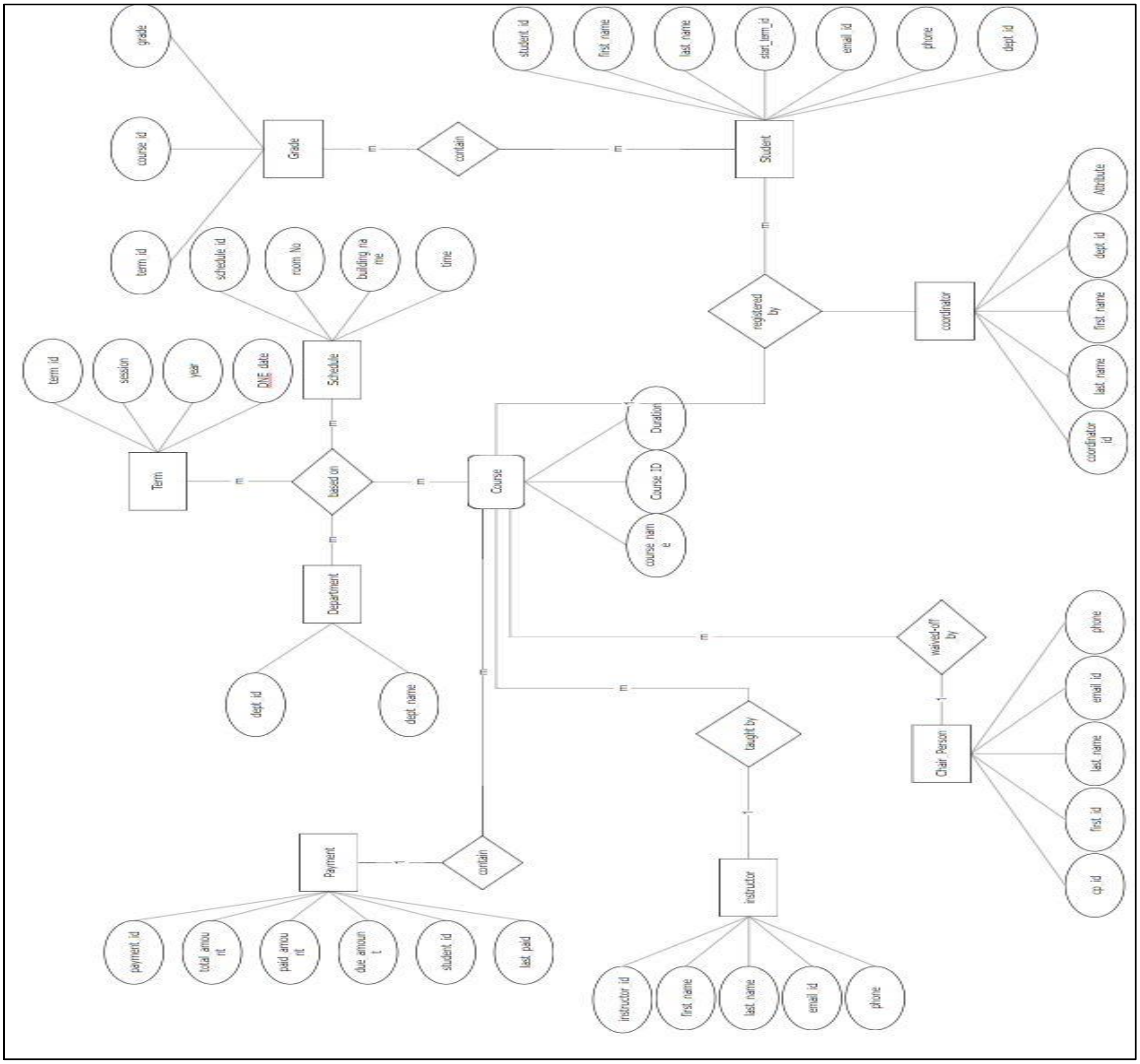


Figure 4 Entity Relationship diagram used in MiniSIS

2. System Model

Finite State Machine(FSM) is used as the model of the Mini SIS. The system can be divided into 4 sub-modules, the module providing course related service for student, the module providing grade related service for student, the module providing course related service for admin, the module providing student related service for admin. We build FSM for the 4 sub-modules respectively.

2.1. FSM of Student Activity

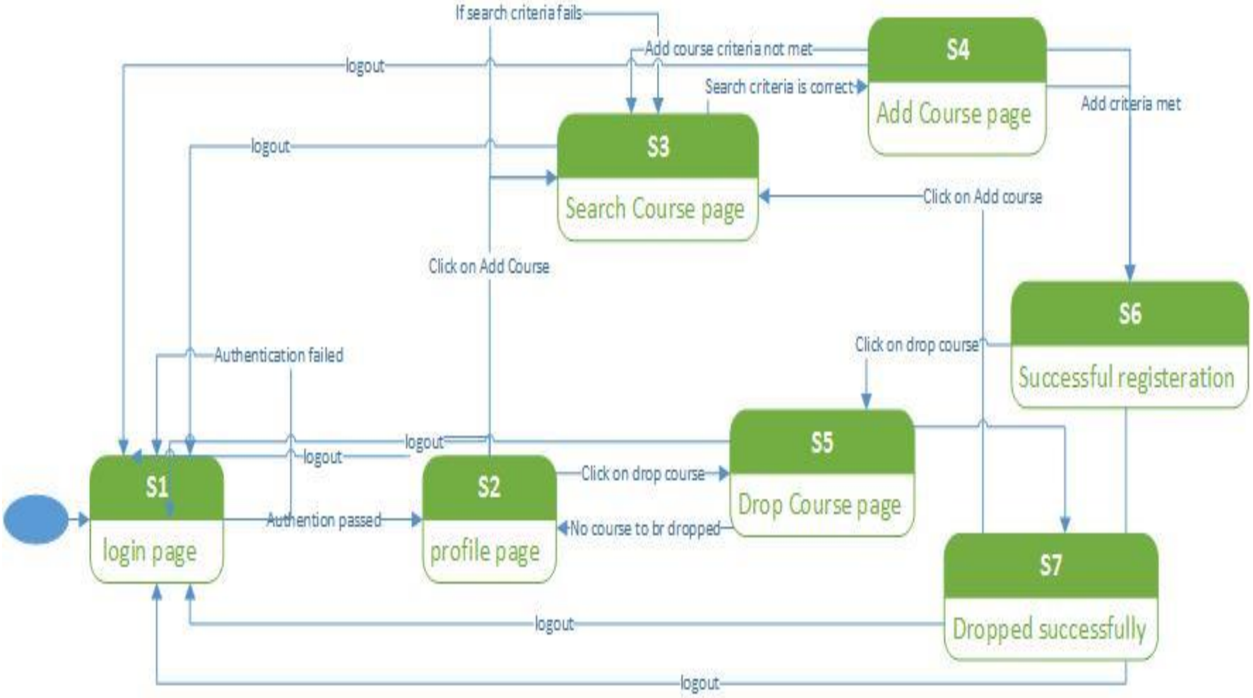


Figure 5 FSM of course related module for student

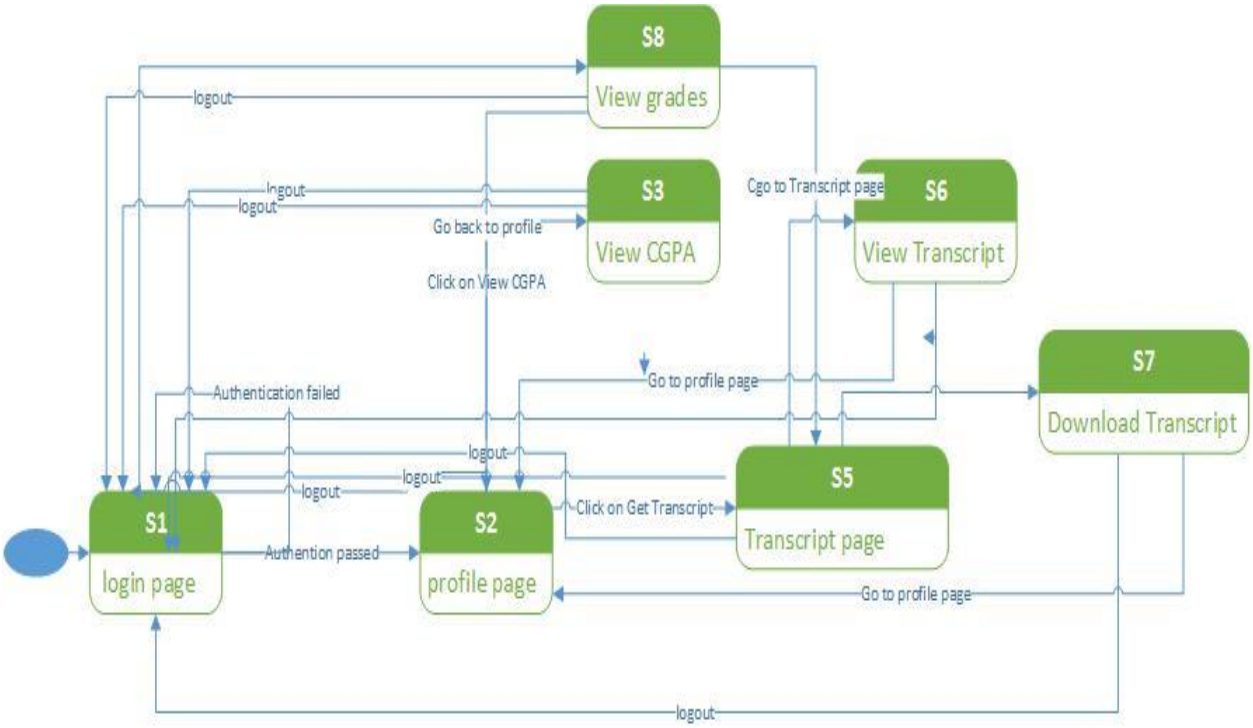


Figure 6 FSM of grade related module for student

2.2. FSM of Admin Activity

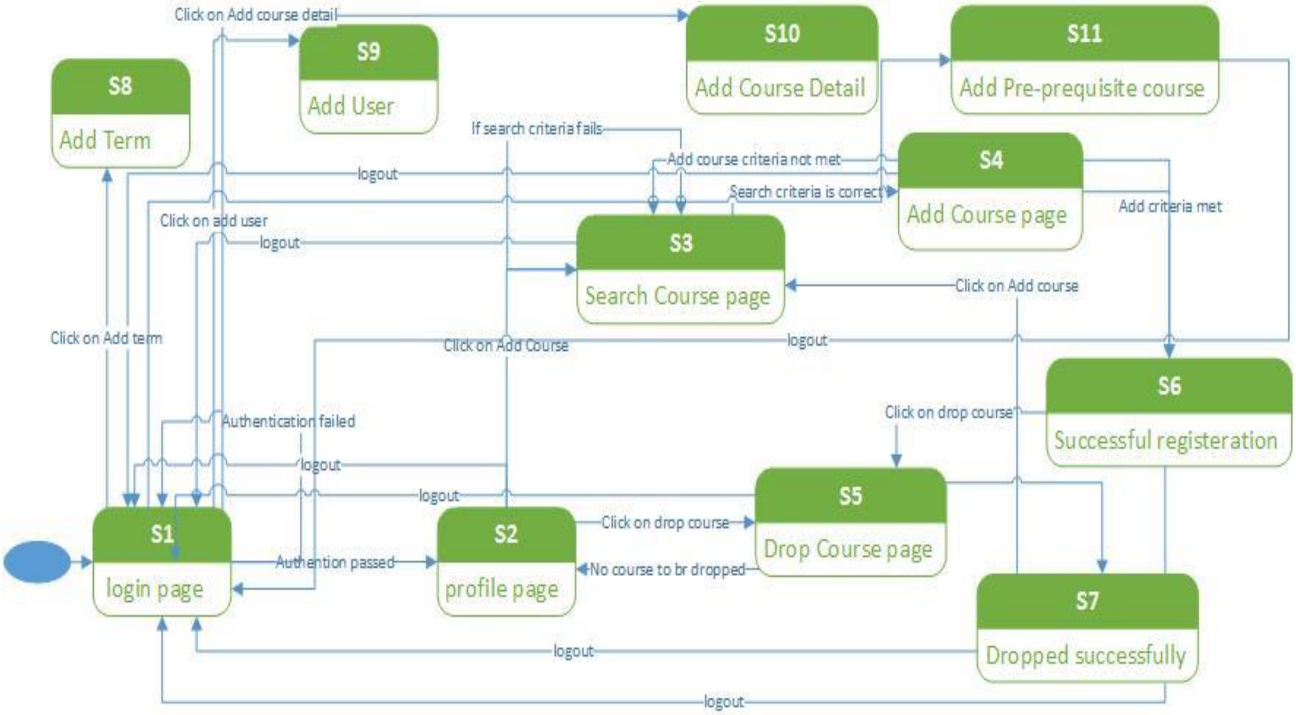


Figure 7 FSM of course related module for administrator

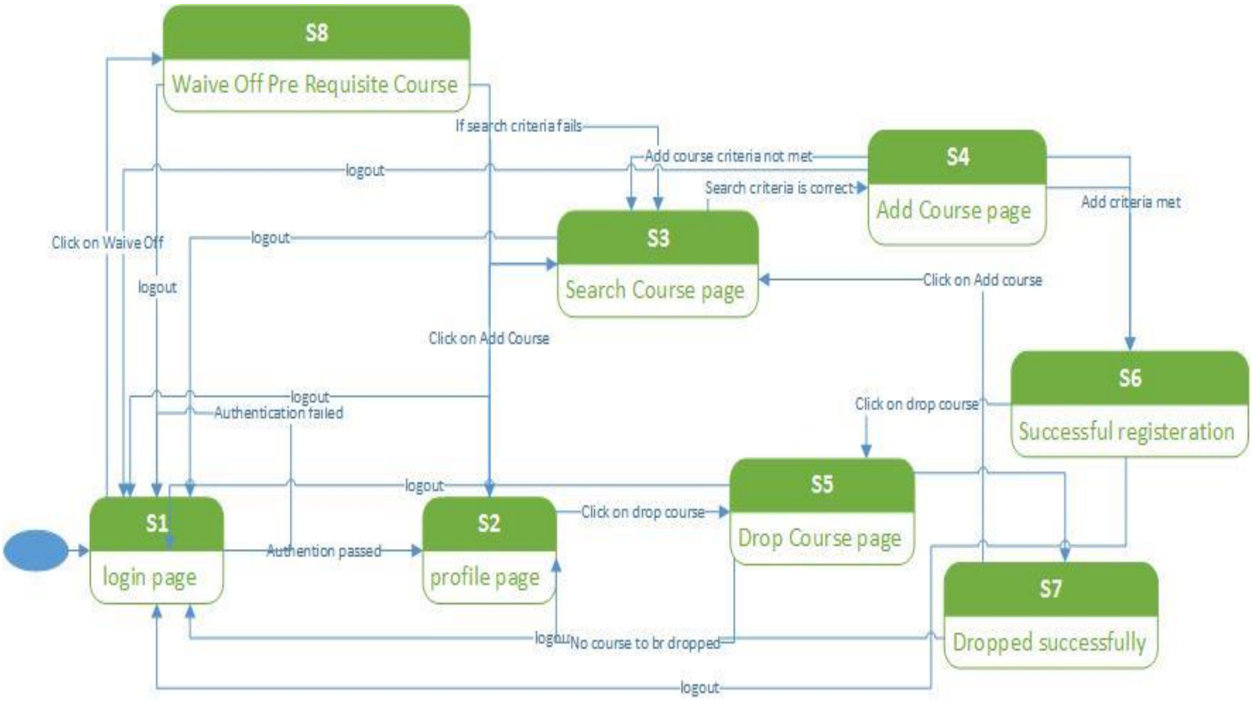


Figure 8 FSM of Course related module for Program Director

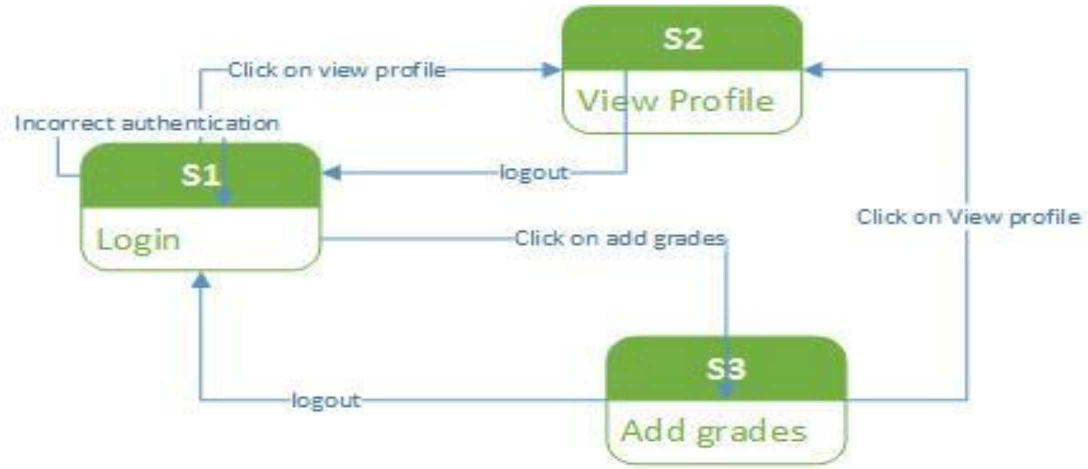


Figure 9 FSM of course related module for Instructor

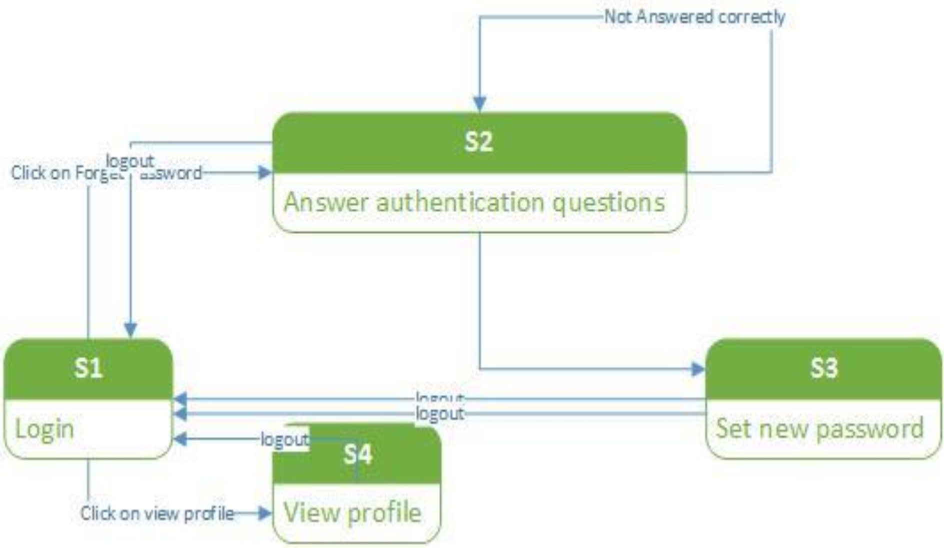


Figure 10 Login module for all users except the admin

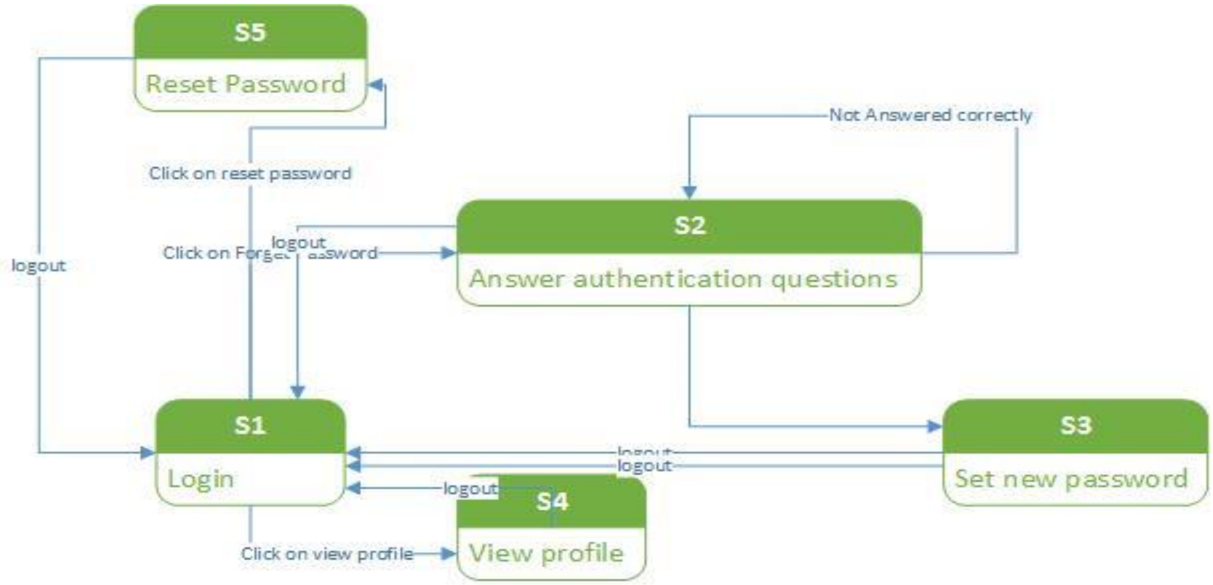


Figure 11 Login module for administrator

2.3. Reachability Analysis

From the FSM of Mini SIS, we can see that all states and transitions can be reached through requests from the user.

3. Inspection Analysis

Checklist for Requirements Specification Reviews

The inspection analysis of SRS given by the QA team is given below.

A – Organization and Completeness

	Review Items	**Yes/ No**
1	Are all internal cross-references to other requirements correct?	**Yes**
2	Are all requirements written at a consistent and appropriate level of detail?	**Yes**

3	Do the requirements provide an adequate basis for design?	**Yes**
4	Is the implementation priority of each requirement included?	**Yes**
5	Are all external hardware, software, and communication interfaces defined?	**No**
6	Does the specification include all of the known customer or system needs?	**No**
7	Is the expected behavior documented for all anticipated error conditions?	**Yes**

Comments:

- System interfaces are missing, the document doesn't have any mention of a payment gateway or third-party, it is a required interface. The defined system interfaces are described as technical constraints rather than what is needed for the system in terms of interfaces
- Hardware interface, it is a redundant description of system interface
- The document did not mention the calendar which is a required feature to add/display important school's dates and events
- The document failed to mention that the system should include features to process payments using credit cards or direct bank deposits. Penalties or interests on late payments is also required for the system to handle but the document has no reference of it

Inspection Analysis:

A – Organization and Completeness

5. Are all external hardware, software, and communication interfaces defined:

We defined the all required information in SRS document senction 2.2 production prospective. Where we mentioned that as per language the software can be installed in any platform and no memory constraints.

6. Does the specification include all of the known customer or system needs?

We have defined in the SRS the stakeholder detail at what level what role and responsibility which user has at section 2.7 and at section 3.5.3. in the SRS document

B- Correctness

	Review Items	**Yes/ No**
1	Do any requirements conflict with or duplicate other requirements?	**No**
2	Is each requirement verifiable by testing, demonstration, review, or analysis?	**Yes**
3	Is each requirement written in clear, concise, unambiguous language?	**No**
4	Is each requirement in scope for the project?	**Yes**
5	Can all of the requirements be implemented within known constraints?	**Yes**
6	Are any specified error messages unique and meaningful?	**Yes**
7	Is each requirement free from content and grammatical errors?	**Yes**

Comments:

- User roles or types are not clear, the document mentions student and admin types only, where the system would need more than this. There are a lot more user types in such as instructor, advisor, and faculty member
- It is unclear what is meant by different iterations when the system is supposed to be delivered in one release, and so some of the important features are scheduled for the second iteration which makes the system incomplete for delivery
- There are no specific requirements about the user interface of the system, no colors, fonts, styles or any other specifications for the look and feel of the interface
- In the specific requirements section, backend requirements, there is a typo which meant to be relational database not rational database
- Assumptions section failed to mention that the system should have some data in it to operate, like departments, programs, courses, as there is no mention of any requirements that will allow admin users to add this data, it should be assumed to exist in the system

Inspection Analysis:

B- Correctness

3. Is each requirement written in clear, concise, unambiguous language?

We assumed on the basis of backend work of university which somehow as per the requires is not at priority level. Still we have developed few features by which admin can manage that work which is not concerned in the given requirement. Based on this constraint we planned to deliver the given features on time.

C- Quality Attributes

	Review Items	**Yes/ No**
1	Are all performance objectives properly specified?	**No**
2	Are all security and safety considerations properly specified?	**No**
3	Are other pertinent quality attribute goals explicitly documented and quantified, with the acceptable tradeoffs specified?	**No**
4	Is each software functional requirement traceable to a higher-level requirement (e.g., system requirement, use case)?	**Yes**
5	Is each requirement uniquely and correctly identified?	**Yes**
6	Have internationalization issues been adequately addressed	**No**
7	Are all time-critical functions identified, and timing criteria specified for	**No**

	them	

Comments:

- Performance requirements are not specific, there is no mention of what is an acceptable or required response time or operating time of the system
- The mention of “0% course conflict” in the performance requirements is irrelevant to performance, it should be mentioned in specific requirements section
- Security requirements are unclear, no mention of what security standards are required or hashing algorithms
- Usability requirements are described as a solution not as how the system is supposed to support usability
- There is no mention of specific quantifiable attributes of the system
- It is unclear if the system should support international standards, different language, or different time zones.

Inspection Analysis:

C- Quality Attributes

In the section 3.6 we mentioned the quality attributes and we have implemented our features accordingly such as avoiding the confliction,password encryption, enrolling management ,and payment handling .

4. Acceptance Test Plan

4.1Goal of the Test Plan

The goal of this Test Plan (TP) for Acceptance Testing is to inform about the activities that are involved in the testing process, the test process involved in it and the deliverables concerning acceptance test plan for Mini Student Information System.

4.2 Assignment

4.2.1 Client

Here Concordia University is a client who has requested an acceptance test plan for the testing phase. This Acceptance plan will be used by the QA team for the Concordia University to run tests on the behalf of Concordia University's customer.

4.2.2 Supplier

Quality assurance team for Concordia University supplies the acceptance test plan and is responsible for executing all test cases and checking the outputs.

4.2.3 Assignment

This acceptance test plan acts as a guideline for QA team to create a report of all the results of test cases executed on the system. All observed and expected outputs are enlisted in this document. It is used to verify if the system presented follows or abides by all the requirements enlisted in the SRS document.

4.3 Scope

The system provides the services based on different classes of users. Those different classes will have different access levels to the system and its features. System administrator will have set of features to manage and maintain accounts of other users, while a faculty member will have access to the faculty calendar management, and course management area. Student account will let the user register or drop courses based on the defined rules of the faculty, see grades and transcripts, and pay tuition fees.
The system to be implemented will not support transferring students from a different school or carrying credits of transferred students between different departments.

4.3.1 Within scope:

The acceptance test plan would test all the basic enlisted in the SRS for the Student Information System. Checks on all the functional and nonfunctional requirements as discussed in section 3.1 are in the scope of this document.

4.3.2 Out of scope:

Unit testing, module testing and integration testing are not within the scope of this document that is responsibly of development team. Also, due to insufficient time to check, nonfunctional requirements as Performance, Maintainability, Portability cannot be checked.

4.4 Preconditions and assumptions

The following demands apply to the test process:

- ➢ Acceptance test plan in a part of total quality approach to Student Information System as mentioned in project plan.

To make the test process successful the following things need to be arranged:

- ➢ The developers should be well acquainted with the databases used by the University.
- ➢ Concordia University should clearly mention their expectations form the software.

- The testing strategy used in the acceptance testing should be in accordance to Concordia University's demands.
- The Project manager should make sure that the development team gets the required access to the database. Also, all the technical documents required for the project are made available.
- Unit testing and Integration testing are taken care by the development team and the report is provided to the QA team.

4.5 Acceptants and acceptance criteria

4.5.1 Acceptants

Acceptants on behalf of the commissioning organization are:

Name	Function	Department
DR RACHIDA DSSOULI	Project manager at Concordia University	Academic Division

4.5.2 Acceptance criteria

The acceptance criteria for system and acceptance tests are:

Description	Norm
It is expected that all the test cases pass with a certain output range and criteria mentioned in section 4.4. All the observed outputs are same as expected outputs	Observed output = Expected output
In case where test cases are expected to fail then suitable error messages or outputs are displayed as mentioned in section 4.4 observed output or error message is same as expected output or error message	Observed output or error message = Expected output or error message
Proper error messages should be displayed in case when an un-expected input is encountered by the system.	Test cases should test system on un expected inputs.

All the non-functional requirements mentioned in SRS document should by satisfid by the test cases. Non-Functional requirements such as performance, reliability, correctness, and other standards that were agreed to in SRS.	All the non-functional requirements that are mentioned in SRS of Concordia University must pass the test cases
Defects mentioned in test report must be fixed by development team.	The resolved defects should be verified by the QA Team

4.5.3 Moment of release

The production phase of the project depends on the test report which is generated by executing the test plan as mentioned in acceptance test document. In case all requirements are met, the product is released.

4.5.3.1 Documentation

This chapter describes the documentation used in relation to the development tests, often the same as the development basis. The documentation that has a relation with the master test plan will also be described in detail.

4.5.3.2 Basis for the test plan

The following documents were used as a basis for this test plan.

Document name	Version	Date(mm/dd/yyyy)	Author
Software Requirement Specification for MiniSIS	1.0	09/26/2017	Development Team

4.5.3.3 Test basis

The test basis consists of the documents from which the test cases are being derived. It contains the documentation that serves as a basis for the tests to be executed. The overview below states the test basis for the system and acceptance tests.

Document name	Version	Date	Author
Software Requirement Specification for MiniSIS	1.0	09/26/2017	Development Team

4.5.3.4 Test strategy

Taking into consideration the time constraint everything cannot be tested thoroughly. Therefore we have decide the task priorities for the test cases. These task priorities will decide the test capacity as effective as possible. The Test Plan document for MiniSIS consists the details of different task priorities in the test strategy section.

The test strategy determines how, when and with what thoroughness is the task should be tested. The motive of test strategy is to find the most critical defects in early stage of process al the least cost.

Execution of risk analysis is the first step in determining the test strategy in the Test Plan. The results of that Product Risk Analyses (PRA) that are relevant for the Acceptance test plan can be found in appendix 1. In this test plan the test strategy from the master test plan has been further elaborated for Acceptance test plan.

The test cases are enlisted in a document that accompanies this document. The traceability of test cases to the characteristics of software mentioned in SRS are given in the Table below.

4.6 Approach

This section describes *how* the testing is handled in conformity with the test strategy

Testing Levels for MiniSIS

Test level	**Goal**	**Responsibility**
Unit Testing	For testing a unit of code while development process is going on.	Development Team
Integration Testing	For testing integration of various modules after they are integrated and made into software.	Development Team
System Test	For testing the system for all functional and non-functional requirements that are agreed to in SRS document.	QA Team
Functional Testing	Reviewing the functionality of the System for end users	QA Team
Acceptance test plan	To test for errors in functionality and usability errors	QA Team

4.7 Description test approach of acceptance test plan

4.7.1 Intake test object

The acceptance test plan starts with the execution of an intake of the test object in order to verify that the entry criteria are met. This intake consists of a completeness check and a pre-test.

4.7.2 Completeness check

The best way to ensure completeness check is to create a checklist of all the test objectives with a list of all the required documents.

4.7.3 Pre-test

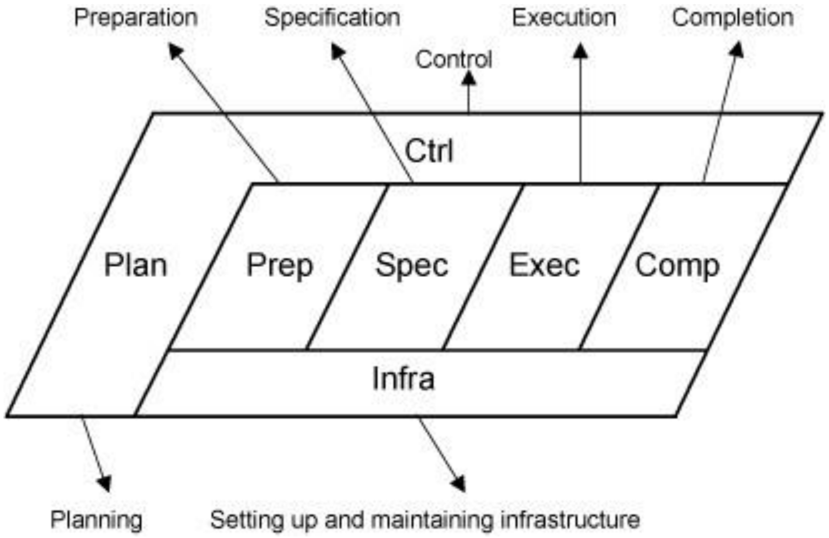
After installation of the test object, a pre-test takes place in order to determine whether the test object is good enough to start testing. The pre-test will be executed as follows:

1. Check whether all the basic functionality of MiniSIS that is adding courses, deleting courses, viewing transcript, paying online etc.
2. Secondly, for all the above functionalities, test cases should be developed to test all the possible scenarios that is success scenario and alternate or failure scenario. These also be tested on all the boundary values.
3. Finally, the design for test cases should be done keeping in mind the functionality and usability from end users point of view.

4.7.4 Acceptance Test Plan

- Week 1: Using a black box approach the system administrator module will be tested for defects such as links that do not work and missing links. All functions will be tested with correct and incorrect input.
- Week 2: The same procedure as week 1 but using a white box approach. Input will be given to all functions with the intention of calling internal functions.
- Week 3: Static testing. The code will be verified to look for uninitialized variables and other invalid code.
- Week 4: Verification that the module meets the system requirements specification and the system design specification.

4.7.5 Phase Acceptance Testing



In the **Planning** phase, the test manager provides a plan that is supported by the client to adequately execute the test assignment. This is laid down in the test plan. In the **Control** phase the activities in the test plan are executed, monitored, and adjusted if necessary. The **Setting up and**

maintaining infrastructure phase aims to provide the required test infrastructure that is used in the various phases and activities. The **Preparation** phase aims to have access to a test basis, The SRS document, agreed to with Concordia University who is client, should make sure that we have test cases of adequate quality and expectations for the MiniSIS.

The tests are specified in the **Specification** phase and executed in the **Execution** phase. This provides insight in the quality of the test object. The test assignment is concluded in the **Completion** phase. This phase offers the opportunity to learn lessons from experiences gained in the project. Furthermore activities are executed to guarantee reuse of products [1].

4.7.6 Entry and exit criteria

The following entrance criteria are defined for the phase Specification and Execution:

Entrance criteria for the Specification phase:

- All the test basis documents that are used for creating acceptance test plan have been shown and approved by Concordia University before creating test cases.
- All initial test phases namely unit, Integration and module testing is carried out properly and all defects that were found are fixed. All the documentation that is reports that were generated for these tests is submitted.
- Final test plan should be approved by Concordia University.

Entrance criteria for the Execution phase:

- All initial test phases namely unit, Integration and module testing is carried out properly and all defects that were found are fixed. All the documentation that is reports that were generated for these tests is submitted
- All the test scripts for acceptance test plan are submitted and are approved by Concordia University.
- The environment for testing is set in accordance with requirements for infrastructure, functionality and test data sets.
- There are no defects that are pending from initial testing phases unit module and integration that may hamper acceptance test plan. These defects will have such an impact on MiniSIS that no operational line, technical processes, administrative processes can be tested.

The following exit criteria have been defined for the user acceptance test phase:

- Test results or reports have been submitted for acceptance test plan and they are verified at QA manager level and by project manager of Concordia University.
- There are no functional or non-functional defects pending in the MiniSIS that may hamper the system to go to next phase that may be production or addition of any new functionality.
- The defects that were found during acceptance test plan have been reworked by development team and are fixed and them QA team has again executed test cases and checked if rework has affected the system positively or not.

4.7.7 Infrastructure

4.7.7.1 Test Environment

Necessary test environment(s): Below is the list of necessary hardware, software test environments.

- The client must have installed the Java version 5 or 6.

- The transaction database at the University end must be available for the software features to work properly.

4.7.7.2 Office setup

Components	**Comment**
Hardware	
1PC with Windows XP or higher	To test the MiniSIS functionalities
1 Backup system	In case of failure.
Workspace	
Computer Lab	To ensure the testing without any external interface.
Small Office	For any project level meeting.
Built-In Packages	

Microsoft office suite	For creation of test report and other documents.
Others	
Refreshment	For motivating the testers

4.7.7.3 Defect Procedure

The unit test is performed on all the major functionalities like:

- View Course Details.
- Add course.
- Drop course.
- View grades.
- Admin can search student and course.
- Admin Add or Drops course.
- Admin updates student grades.

These test cases and test scenarios must be validated and verified. All defects must be reported to development team for bug fix and recorded by QA team.

5. Acceptance Test Suite

5.1. Test case 1

Tested By	QA Team
Test Type	Functional (Unit)
Test Case Number	TC01
Test Case Name	Login
Test Case Description	The user will get error message after entering wrong user name or password
Item(s) to be tested	

1.the login page should be functional 2.Username and password are authenticable or not 3.validating that the user role is as per the username or not	
Specification	
Input	**Output**
1.In the main page click on the login button	You will get the access to the portal if the username and password are validated
2. Wrong Password to be entered	Error message to be displayed
Procedural Steps	
1. Entering the user Id and password by user 2. Authentication the user by system 3. Showing error message to the user after entering the wrong characters as user Id or password	
Test Results: Pass	

5.2. Test case 2

Tested By	QA Team
Test Type	Functional (Unit)
Test Case Number	TC02
Test Case Name	View profile
Test Case Description	The user wants to view her/his profile
Item(s) to be tested	
1. Using correct password and username 2. Login to the system 3. View her/his profile with her/his individual information	
Specification	
Input	**Output**
1.validated Username and password	The profile will b e opened
Procedural Steps	

1. User open the main page of the university 2. User select the login page 3. User use the username and password 4. The username and password verification activated 5. User login to her/his profile
Test Results: Pass

5.3. Test case 3

Tested By	QA Team
Test Type	Functional (Unit)
Test Case Number	TC03
Test Case Name	Change password
Test Case Description	User apply for changing password
Item(s) to be tested	
1. Changing password option	
Specification	
Input	**Output**
1.Old password	If the old password is correct the system accept it
2.security question	Security question accepted
3.new password	A new password with acceptable charactrs
Procedural Steps	
1. User select the change password option 2. The page for changing password will open 3. The user should enter her/his old password 4. The user should select one security question 5. The user should enter the answer for security question	

6. The user should enter new password with acceptable characters
Test Results: Pass

5.4. Test case 4

Tested By	QA Team
Test Type	Functional (Unit)
Test Case Number	TC04
Test Case Name	Forgot password
Test Case Description	Using wrong password
Item(s) to be tested	
1.User cannot login to the system using wrong password 2.Error message should be appeared	
Specification	
Input	**Output**
1. Wrong password	Error message
Procedural Steps	
1. User enter the username and password 2. The error message will appeared that the password is wrong 3. The user enter the correct password or select the forgot password option	
Test Results: Pass	

5.5. Test case 5

Tested By	QA Team
Test Type	Functional (Unit)
Test Case Number	TC05

Test Case Name	Course Search as Student
Test Case Description	Test the functionality of searching for courses by department and course name/code as search criteria
Item(s) to be tested	
Course search page and search results	
Specification	
Input	**Output**
Department and Course name or code	List of course that match the criteria of selected department and course name/code
Procedural Steps	
1. Login to the system using a student account 2. Navigate to Course Search 3. Select a department from the list of departments 4. Type course name or code in the box 5. Click Search	
Test Results: Pass	

5.6. Test case 6

Tested By	QA Team
Test Type	Functional (Unit)
Test Case Number	TC06
Test Case Name	Register in Course using Student account
Test Case Description	Test Register in course functionality
Item(s) to be tested	
Course registration page/section	
Specification	

Input	Output
1.Login to the system using student Id	Access to the profile
2.opening the list of the course	Course appears on student list of registered courses
3.Selecting the Course	The course selected
4.Final registration	The student registered in the course
Procedural Steps	
1. Login to the system using student account 2. Navigate to Course Registration 3. Search for a course with the following criteria: a. Student was not registered in the course before b. Course is from the same department of the student c. Course is open for registration d. Course does not have prerequisites that the student does not qualify for 4. Select the available course for registration 5. Select enroll in course	
Test Results: Pass	

5.7. Test case 7

Tested By	QA Team
Test Type	Functional (Unit)
Test Case Number	TC07
Test Case Name	Drop a Registered Course using Student account
Test Case Description	Test dropping a registered course from list of registered courses
Item(s) to be tested	
Course registration page/section	

Specification	
Input	**Output**
Student Account with Course ID	Course disappears on student list of registered courses
Procedural Steps	
1. Login to the system using student account which has some courses registered 2. Navigate to Course Registration 3. Enter the student ID of an existing student 4. Search for a course with the following criteria: a. Student was not registered in the course before b. Course is from the same department of the student c. Course is open for registration d. Course does not have prerequisites that the student does not qualify for 5. Select the available course for registration 6. Select enroll in course	
Test Results: Pass	

5.8. Test case 8

Tested By	QA Team
Test Type	Functional (Unit)
Test Case Number	TC08
Test Case Name	View Unofficial Transcript
Test Case Description	The Unofficial transcript acceptable to student
Item(s) to be tested	
1. verified that report page is active for both users 2.The transcript should be accessible for student	
Specification	

Input	**Output**
1. Select report page	Report page should get displayed.
2. Select view transcript	Transcript should be viewed
Procedural Steps	
1. Student should be logged in to the system 2. Student should select the report page 3. It should be redirect on transcript page 4. Click on view transcript button 5. The transcript will appeared to the student	
Test Results: Pass	

5.9. Test case 9

Tested By	QA Team
Test Type	Functional (Unit)
Test Case Number	TC09
Test Case Name	View CGPA
Test Case Description	The student wants to check the CGPA
Item(s) to be tested	
1. The function to view the CGPA	
Specification	
Input	**Output**
1. Student select the view CGPA option	The CGPA will appear
Procedural Steps	
1. The student log in to the system using correct username and password	

2. The student select the Activity menu 3. Student select the view CGPA option 4. CGPA will appear
Test Results: Pass

5.10. Test case 10

Tested By	QA Team
Test Type	Functional (Unit)
Test Case Number	TC10
Test Case Name	View CGPA
Test Case Description	The admin wants to check the CGPA
Item(s) to be tested	
2. The function to view the CGPA	
Specification	
Input	**Output**
1. Admin select the view CGPA option	System ask for student information (only student id is enough)
Procedural Steps	
1. The Admin log in to the system using correct username and password 2. The admin select the Activity menu 3. Admin select the view CGPA option 4. The admin enter the student Id number 5. CGPA will appear	
Test Results: Pass	

5.11. Test case 11

Tested By	QA Team
Test Type	Functional (Unit)
Test Case Number	TC11
Test Case Name	Pay Fee
Test Case Description	To verify that the tuition fee has been paid
Item(s) to be tested	
1. The tuition fee needs to be paid 2. The amount is paid successfully 3. Some amount is pending or not 4. The pending amount will be changed after paying the tuition or not	
Specification	
Input	**Output**
1. Select the payment page	Payment page should be opened and displayed
2. Select the pay option	Enter the payment information
3. Click the submission	Success message will be displayed
4. Due date	Error message will be shown after due time
Procedural Steps	
1. User login to the system 2. Select the payment page 3. Select the payment option 4. Enter the required amount 5. Submit the payment	
Test Results: Pass	

5.12. Test case 12

Tested By	QA Team
Test Type	Functional (Unit)

Test Case Number	TC12
Test Case Name	View Pre-requisite
Test Case Description	Add pre-requisite course detail
Item(s) to be tested	
1. The student detail should be added successfully 2. The pre-requisite should be added if it is needed	
Specification	
Input	**Output**
1. Admin enter the student Id	The student information should be appeared
2. The enter the course description	The pre-requisite should be added successfully
Procedural Steps	
1. Admin login to the system 2. Admin select the Add Pre-requisite Course Detail 3. Admin enter the student ID 4. Admin select the program and the course name	
Test Results: Pass	

5.13. Test case 13

Tested By	QA Team
Test Type	Functional (Unit)
Test Case Number	TC13
Test Case Name	Add Grade
Test Case Description	
Item(s) to be tested	
1.The instructor must can add the grades of students	
Specification	

Input	Output
The grades of students	Student can see their grades in their profiles
Procedural Steps	
1. Instructor enter the username and password 2. Instructor login to her/his profile 3. Instructor goes to add the grades 4. Instructor upload the grades **5.** Instructor submit the grades	
Test Results: Pass	

5.14. Test case 14

Tested By	**QA Team**
Test Type	**Functional (Unit)**
Test Case Number	TC14
Test Case Name	Logout
Test Case Description	User wants to exit from the portal
Item(s) to be tested	
System Logout	
Specification	
Input	**Output**
1. Using the logout feature(button) by user	The user's profile become logged out
Procedural Steps	
1. User's profile goes to log out **2.** System goes back to login window	
Test Results: Pass	

6. Installation Instructions

6.1. Environment configuration

Installations Required:

- JDK
- Eclipse with JavaFX
- MYSQL
- MySQL Workbench

6.2. Data schema

The data schema MiniSIS should be imported in MySQL Workbench.

7. Features implementation

As per the SRS , the following features have been implemented:

7.1 Implemented features:

User Login
Search Course
Add Courses
Course Drop
View Schedule
View Unofficial Transcript
View CGPA
Pay Fee
Waive off Pre-requisite course
View Profile
Add Grades
Logout
View Due Amount
Pay by admin

View Grades
Add Grades
View profile
Change Password

In payment feature, no fake payment system is used as mentioned in the SRS.

7.2 Extra features Implemented:

Forget password
Set security Question
Reset Password
Add Users
Add course Detail
Add Term
Add Pre-Requisite Course Detail

8. Unit Testing

Unit testing covers the following features:

- Add Course
- Drop Course
- Waive Off Pre-requisite course
- View CGPA
- Pay Fee